



How do I start contributing to Yocto Project?

Michael Opdenacker, Bootlin

Yocto Project Summit, 2022.05



But the first question is...

What shall I contribute to the Yocto Project?

We need you, you, you

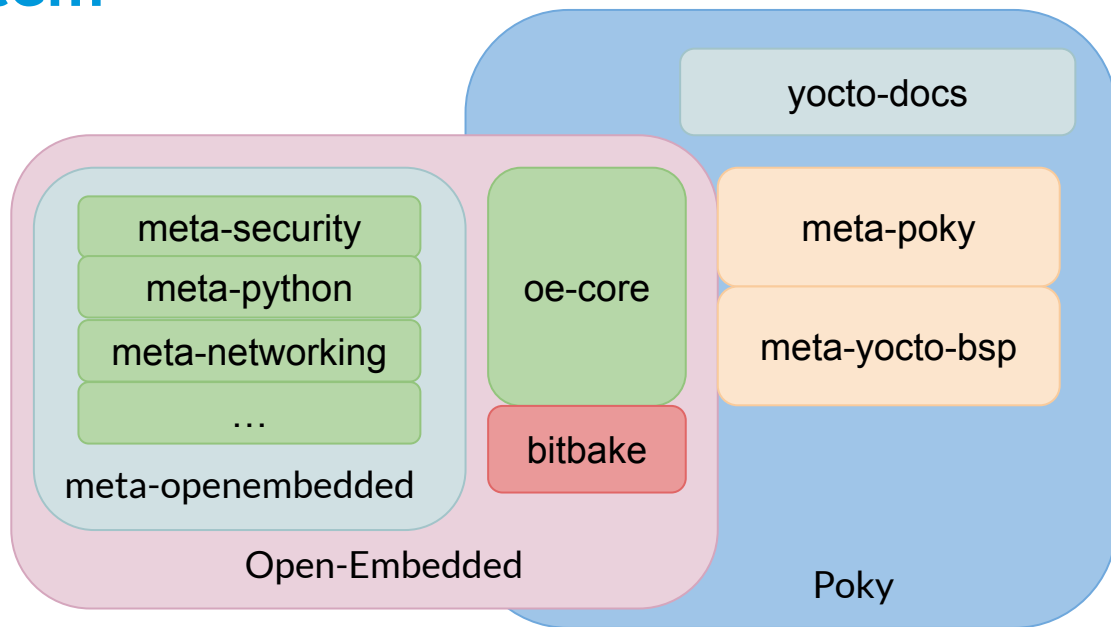
- Your interests
- Your experience
- Your ideas
- Your personality

Our greatest value is our community members



<https://openclipart.org/detail/262931/prismatic-brain-word-cloud>

Yocto Project ecosystem



- **OpenEmbedded:** build system
- **BitBake:** build engine
- **Poky:** reference distribution

The Yocto Project

Join mailing lists and IRC

- **Mailing lists:**
 - <https://www.yoctoproject.org/community/mailling-lists/>
 - Most important ones: [Yocto](#), [Yocto-announce](#), [OE](#), [OE-core](#), [BitBake Developers](#), [Docs](#)
- **IRC: #yocto and #oe on Libera.chat**
 - With your favorite IRC client
 - Using a Matrix client

Read the documentation and contribute to it

- A good way to get started with the Yocto Project
- Try to run the provided examples. Some of them can be out of date, but we are not aware of this.
- Report typos and suggest improvements
- Share your experience and propose new content
- Get familiar with Sphinx, great for creating your own documentation.
- Review posts

Supporting users

The main strength of the Yocto Project is its community

- Support users on the [Yocto mailing list](#)
- Share your knowledge on IRC too

Get your hands dirty - Help with newcomer bugs

Newcomer Bugs

ID	Summary (24 tasks) →	Assignee	Status	Milestone	Severity	P	Whiteboard
3362	<i>Improve dependency analysis tools</i>	New Comer Bugs	NEW	4.99	enhancement	Medium+	NEWCOMER
5725	Add QA check for installing Python Eggs	New Comer Bugs	NEW	4.99	enhancement	Medium	NEWCOMER
6707	Add ptest for curl	Unassigned	NEW	Future	enhancement	Medium	NEWCOMER
6803	Sanity check for non-largefile use	New Comer Bugs	NEW	4.99	enhancement	Medium	NEWCOMER
8222	Elide dependency from -dev package on main package if there are no shared libraries in it	New Comer Bugs	NEW	4.99	enhancement	Medium	NEWCOMER
8744	Need to standardize on a format for security bugs	New Comer Bugs	NEW	Q4	normal	Medium	NEWCOMER
9070	Add test that locales are working	New Comer Bugs	ACCEPTED	4.99	enhancement	Medium	NEWCOMER
9451	Document a cookbook procedure on how to change kernel version	Unassigned	IN PROGRESS REVIEW	Future	normal	Low	NEWCOMER
9692	QA check for #! interpreter should handle '/usr/bin/env something'	New Comer Bugs	NEW	Future	normal	Medium	NEWCOMER
10165	<i>Check that the locale en_US.UTF-8 is available</i>	Unassigned	NEW	4.99	normal	Medium+	NEWCOMER
10266	Add ptest for gnutls	Unassigned	NEW	4.99	enhancement	Medium	NEWCOMER
10636	Improved image testing by extending automated QA tests	New Comer Bugs	NEW	4.1	enhancement	Medium	NEWCOMER
11053	<i>insane.bbclass: Verify shebang line length valid</i>	Unassigned	NEW	Future	enhancement	Medium	NEWCOMER
11142	Use artificial recipes for oe-selftest where possible to reduce work	New Comer Bugs	NEW	4.99	enhancement	Low	NEWCOMER
11394	<i>OEQA devtool - Add tests to cover devtool handling of various git URL styles</i>	New Comer Bugs	NEW	4.99	enhancement	Medium+	NEWCOMER
11605	bitbake documentation: "file-checksums" needs to be documented	New Comer Bugs	NEW	4.99	enhancement	Medium	NEWCOMER
12389	oe-selftest devtool tests fails when repository comes from a tarball	New Comer Bugs	NEW	4.99	normal	Medium	NEWCOMER
12716	Replace unfs3 with nfs-ganesha?	New Comer Bugs	NEW	Future	enhancement	Medium	NEWCOMER
12904	Port panel from apm/acpi to sysfs	Unassigned	NEW	4.1	enhancement	Medium	newcomer
13425	Add bblock and bblock helper tools	New Comer Bugs	NEW	4.1 M1	normal	Medium+	NEWCOMER
13819	<i>No clue when CONNECTIVITY_CHECK_URI is inaccessible</i>	New Comer Bugs	NEW	4.1 M1	minor	Medium+	NEWCOMER
13846	<i>Add runtime test for Piglit</i>	Unassigned	NEW	4.99	enhancement	Medium+	NEWCOMER
13996	<i>Add selftest for gdbserver</i>	Unassigned	NEW	4.1	enhancement	Medium	NEWCOMER
14446	<i>devtool add malformed PACKAGECONFIG</i>	Unassigned	NEW	4.1 M1	normal	Medium+	NEWCOMER
24							

[https://wiki.yoctoproject.org/wiki/Bug_Triage#Newcomer Bugs](https://wiki.yoctoproject.org/wiki/Bug_Triage#Newcomer_Bugs)

Participate to code reviews

- Reviewing patches is a good way to learn
- It also helps to raise the quality of our code
- It really helps maintainers and reduces their load
 - Example: as the documentation maintainer, I owe a lot to Quentin Schulz, doing many reviews of documentation patches.

Patch submission and review process

- The patches should be sent on public mailing lists, and be sent as inline attachments through **git send-email** (guarantees that the patches will be applicable)
- This makes it easy for reviewers to comment and questions by replying to specific lines of the patch.
- This is very different from the processing of Pull Requests in tools such as GitHub.

Preparing patches (1)

- Find the original repository. Typically, for Poky, read <https://git.yoctoproject.org/poky/tree/README.md>
- Create a new branch for your changes
git checkout <ref-branch>
git checkout -b <my-branch>
- Make your changes and test them as much as possible
- Prepare your commits. Each commit should only solve one problem at a time. This makes reviewing easier.

Preparing patches (2)

- **Create each commit**
git add -A <files> (if new files)
git commit -s <files> (with specific files)
or git commit -as (with all modified files)
 - Make sure to include a good commit message (see next page)
 - Don't forget the Signed-off-by info (-s in git commit)
- **Generate patches (one patch per commit)**
git format-patch <ref-branch>

Create a good commit message (1)

In particular the first line

- It could appear in the release notes
- Important for Poky too, aggregating multiple repositories.
Make sure that the first line is prefixed by a keyword which allows to identify the repository it comes from.
- A good idea is to look at previous commits on the same file(s):

```
> git log --oneline bitbake/bin/bitbake-layers
```

```
2b399a01b5 bitbake: bitbake: enable python warnings at the first opportunity  
1a9e0997c8 bitbake: Revert "bitbake-layers: add signal handler to avoid exception"  
600e35678b bitbake: bitbake-layers: Keep loglevel in colored logger  
cf9c0be3f6 bitbake: bitbake: Drop duplicate license boilerplate text  
79834a7144 bitbake: bitbake: Add initial pass of SPDX license headers to source code  
b0f0eaf63c bitbake: tinfoil: add a parse_recipes() function
```

Make a good commit message (2)

- The commit message should be concise and informative, allowing to know what the change is about without having to look at the code.
- Use the present tense to describe what the patch does:
db3f589f49 bitbake: bitbake-layers: **add** commands for adding and removing layers
- Assume the patch is accepted when writing the commit. Don't add questions or alternative solutions. Those belong to the patch review process.

Good and bad commit messages

Even we frequent contributors could do better!

git log --oneline documentation/Makefile

ecab5f3655 docs: add poky.yaml and sphinx-static/switchers.js to "make clean"

Good, we know it's a documentation patch

8d3e9aaede Makefile/set_versions: Allow poky.yaml to be autogenerated

Bad, we don't know it's about the documentation Makefile

6910fceb7a docs: add "make all" to build old types of docs (html, epub, pdf) at once

Good, we know it's a documentation patch

bca3caeaee Makefile: allow epub and latexpdf outputs to use diagrams from SVG sources

Pretty bad, you have to read the summary to guess it's documentation

c2a68819d0 documentation: Makefile: turn warnings into errors by default

Good

ba2e630546 Makefile: enable parallel build

Bad, you have to read the patch contents to know it's a documentation patch.

Sending patches

- Need to send your patches through **git send-email**
- If doing this for the first time, send the patches to yourself, and check the result
`git send-email --to me@domain.tld 00*.patch`
- Check the **README** file in the repository to know which list(s) to send the patches to
- Then send the patches to the list(s). Example:
`git send-email --to bitbake-devel@lists.openembedded.org
--cc docs@lists.yoctoproject.org 00*.patch`
- Take the reviews into account, and resubmit if needed.

Meet other developers

- **Participate to the weekly engineering and bug triage meetings**
 - <https://www.yoctoproject.org/public-virtual-meetings/>
 - Everyone will feel welcome
- **Participate to the OpenEmbedded Happy Hours**
 - https://www.openembedded.org/wiki/Happy_Hours
 - Get notified through the OE and Yocto mailing lists

All events: <https://www.openembedded.org/wiki/Calendar>

Help fix CVE vulnerabilities

Branch: master

New this week: 3 CVEs

CVE-2022-1381 (CVSS3: 7.8 HIGH): vim <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-1381> *
CVE-2022-1420 (CVSS3: 5.5 MEDIUM): vim <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-1420> *
CVE-2022-29458 (CVSS3: 7.1 HIGH): ncurses:ncurses-native <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-29458> *

Removed this week: 4 CVEs

CVE-2015-20107 (CVSS3: 9.8 CRITICAL): python3:python3-native <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2015-20107> *
CVE-2021-28544 (CVSS3: 4.3 MEDIUM): subversion <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-28544> *
CVE-2022-1304 (CVSS3: 7.8 HIGH): e2fsprogs:e2fsprogs-native <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-1304> *
CVE-2022-24070 (CVSS3: 7.5 HIGH): subversion <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-24070> *

Full list: Found 9 unpatched CVEs

CVE-2019-12067 (CVSS3: 6.5 MEDIUM): qemu:qemu-native:qemu-system-native <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2019-12067> *
CVE-2020-18974 (CVSS3: 3.3 LOW): nasm:nasm-native <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-18974> *
CVE-2021-20255 (CVSS3: 5.5 MEDIUM): qemu:qemu-native:qemu-system-native <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-20255> *
CVE-2022-0529 (CVSS3: 7.8 HIGH): unzip:unzip-native <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-0529> *
CVE-2022-0530 (CVSS3: 7.8 HIGH): unzip:unzip-native <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-0530> *
CVE-2022-1210 (CVSS3: 6.5 MEDIUM): tiff <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-1210> *
CVE-2022-1381 (CVSS3: 7.8 HIGH): vim <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-1381> *
CVE-2022-1420 (CVSS3: 5.5 MEDIUM): vim <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-1420> *
CVE-2022-29458 (CVSS3: 7.1 HIGH): ncurses:ncurses-native <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-29458> *

Automatically reported to the yocto-security and OE-core mailing lists
Opportunity to collaborate with upstream too.

Adopt a package

- **Volunteer to maintain package(s)**
 - OE-core maintainers:
<https://git.openembedded.org/openembedded-core/tree/meta/conf/distro/include/maintainers.inc>
 - You also get help from OE's Auto Upgrade Helper
 - A great way to learn about how the project works
- **Maintainers are also needed for parts of the build infrastructure:**
 - <https://git.openembedded.org/openembedded-core/tree/MAINTAINERS.md>

Other contribution ideas

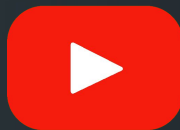
- Read the code (recipes, BitBake) and propose fixes and improvements
- Investigate issues reported by the autobuilder:
<https://errors.yoctoproject.org/Errors/Latest/Autobuilder/>
- Share your experience by proposing a talk at the next Yocto Project Summit
- Contribute to the Yocto Project and OpenEmbedded wikis

Useful resources

- Most YP / OE mailing lists are archived on <https://lore.kernel.org/>
 - Great resource for searching for past messages
 - See <https://www.yoctoproject.org/community/mailling-lists/>
- Further guidelines for contributing patches:
 - https://openembedded.org/wiki/How_to_submit_a_patch_to_OpenEmbedded
 - <https://www.kernel.org/doc/html/latest/process/submitting-patches.html>

Questions?

yocto@lists.yoctoproject.org



yocto
PROJECT

THE
LINUX
FOUNDATION